Evaluation of automated stability testing in machining through closed-loop control and Bayesian machine learning

Jaydeep Karandikar [a],[*],[1], Kyle Saleeby [a],[1], Thomas Feldhausen [a], Thomas Kurfess [a], Tony Schmitz [a],[b], Scott Smith [a]

[a] *Manufacturing Demonstration Facility, Oak Ridge National Laboratory, Knoxville, TN 37932, USA*
[b] *Department of Mechanical, Aerospace, and Biomedical Engineering, University of Tennessee, Knoxville, Knoxville, TN 37996, USA*

ARTICLE INFO

ABSTRACT

This paper describes a system for automated identification of the optimal stable cutting parameters in milling through Bayesian machine learning and closed-loop control. The closed-loop control system consists of a process monitoring architecture, an analysis framework, and a feedback mechanism. The analysis framework consists of a Bayesian machine learning algorithm that learns a stability map given test results. The learned stability map is used to select parameters for stability testing using an expected improvement in the material removal rate criterion. The test parameters are communicated to the machine controller to complete the test cut through a feedback mechanism. The test cuts were monitored using an audio signal; the stability of the test cut was determined by analyzing the frequency content of the audio signal. The test result was fed back to the Bayesian learning algorithm to complete the loop. Experimental results demonstrate that the system can identify the optimal stable parameters without information about the cutting force model or the structural dynamics. The system provides a low-cost method for optimal stable parameter identification in an industrial environment.

1. Introduction and motivation

High-speed machining is a well-known capability for improving machining productivity. However, the cost-efficient implementation of high-speed machining is hampered by chatter, a self-excited vibration that depends on the structural dynamics and the machining parameters [1]. To avoid chatter, a stability map can be used to select stable machining parameters over a domain including the axial depth of cut and spindle speed [1]. Different methods exist to predict stability diagrams in machining. Earlier work included the.

analytical and the frequency domain methods to predict the stability map, which separates stable and chatter combinations of axial depth of cut and spindle speed [2,3]. Alternative approaches include time-domain simulations, semi-discretization, and time finite element analysis that can predict the period-2 and secondary-Hopf instabilities [4 –7]. Recent advances in stability modeling in machining include the following. Olvera et al. described a homotophy method with simulated annealing to determine the stability boundary and optimal cutting conditions; results showed a fast computation time [8–9]. Yan et al. used the Legendre and Chebyshev

---

* Corresponding author.
E-mail address: karandikarjm@ornl.gov (J. Karandikar).
[1] First authors with equal contributions.

polynomials to approximate the state and the time-delayed term in the equations of motion in milling [10]. Qin et al. described a Chebyshev wavelet-based approach for past prediction of milling stability [11]. Celikag et al. presented the validity of mode coupling chatter for flexible machine tools [12].

Although different models and methods to predict the machining stability map exists, there are three main hurdles to implementation in an industrial environment for process optimization. First, stability models need information on the structural dynamics, typically represented by the tool tip frequency response function (FRF), and the cutting force model, which relates the cutting force required to shear away the work material to the commanded machining parameters. Small- and medium-sized manufacturers may not have the required equipment (such as an instrumented hammer and vibration transducer) and the engineering support to complete the FRF measurements. Accurate identification of the cutting force model coefficients requires the measurement of cutting forces for a given tool-material combination with a cutting force dynamometer, which poses an additional cost and challenge. Second, the details for special geometry tools, such as the tooth spacing for a variable pitch end mill, may not be available and they are required for modeling purposes. Third, there exists uncertainty in analytical model predictions due to uncertainty in the inputs, or factors that are not considered in the model [13,14]. As a result, experiments are often required to verify the predicted stability map and the optimal stable parameters.

To address the challenges of implementing the stability map to maximize material removal rates in an industrial environment, a closed-loop control system for automated identification of optimal stable operating parameters is described in this paper. The authors build on prior efforts in Bayesian machine learning for stability map and optimal operating parameter identification in milling by automating the procedure with closed-loop control through communication with the machine [15,16]. Recent developments in communication frameworks between external analytic engines and commercial computer numerical control (CNC) machining centers enable the development and implementation of advanced workflows and operations. Feedback mechanisms have been developed to enable the automatic update of machine instructions, accounting for previous actions, and enabling a closed-loop control regime for standard machining equipment [17].

The closed-loop control system implemented here consists of: 1. an architecture to monitor the machine state through MTConnect; 2. an analysis framework including a Bayesian machine learning method to learn the stability boundary using test results and a strategy to select the test parameters; and 3. a feedback mechanism to communicate the test parameters to the machine controller to complete the test cut. The main contribution of this paper is to demonstrate the combination of Bayesian machine learning and test point selection strategy with closed-loop control for automating the identification of optimal stable operating parameters using a commercially available machine tool controller. The system objective is a low-cost method to quickly identify the optimal stable operating parameters without the requirement of knowledge of the structural dynamics and the cutting force model coefficients.

Research efforts in the literature have focused on chatter control and suppression using machining parameter modification used two different approaches: 1. adjust machining parameters [18–22], and 2. disrupt regeneration of waviness through spindle speed variations [23–28]. In the parameter adjustment method, the goal is to iteratively identify the stable spindle speed by matching the tooth passing frequency to the dominant chatter frequency. Pioneering early work by Smith et al. described a system that automatically detected chatter using the machining audio signal content, stopped the feed when chatter was identified, changed the spindle speed to a new spindle speed based on the chatter frequency, adjusted the feed to keep the chip load constant, and then released the feed hold [18 –20]. Dijk et al. built upon the idea to implement a second control strategy that modifies the spindle speed and feed to reduce the magnitude of tool vibrations [21]. Tarng et al. developed an algorithm to select spindle speed automatically to eliminate chatter in milling based on the features on the cutting-force spectrum [22]. In the spindle speed variation (SSV) method, the regeneration of waviness is interrupted by continuously varying the spindle speed. This modifies the phasing between vibration and surface undulations, which tends to reduce chatter [23–28]. Lin et al. showed that been shown that a sinusoidal variation in spindle speed is the most feasible profile to suppress chatter [23]. Semi-discretization approaches showed improved stability and chatter suppression through spindle speed variation [24 –25]. Zatarain et al. developed a general theory for analysis in the frequency domain and any speed variation strategy [26]. Ding et al. described an active chatter suppression approach through closed loop control by adaptive adjustment of frequency and amplitude of spindle speed variation. [27–28]. Urbikain described a spindle speed variation model using perturbation theory and demonstrated its implementation [29]. Totis et al. presented a method to evaluate process stability in milling with spindle speed variation using the Chebyshev colloation method [30]. A review of prediction methods and experimental techniques is provided in [31].

Earlier methods of CNC control often enabled the transfer of information, but not its execution. CNC control architectures were designed to have minimal access for operational instruction sets, other than standard G-code. Schofield et al. contributed to a more open control architecture; however, it was not widely adopted in industrial applications [32]. In recent years, additional teleoperation of CNC systems was developed by Oliveira et al. but still required user interaction to command the machine [33].

The approach described in this paper is an improvement over those systems that adjust the machining parameters (spindle speed and feed rate) in the following ways. First, the system is designed for pre-process, offline identification of optimal stable machining parameters which maximize the material removal rate. Spindle speed variation methods described in the literature are online methods for eliminating chatter in real-time. Second, the method recommends an axial depth of cut in addition to spindle speed to iteratively converge on the optimal stable operating parameters. As a result, the system can converge to the optimal stable parameters starting from initial stable parameters. Third, the system provides a probabilistic stability map using test results which can be archived and reused for subsequent uses of the tool in the machine.

The remainder of the paper is organized as follows. Section 2 describes the Bayesian learning method for identifying the stability map and the experimental strategy for selecting test parameters. Section 3 describes the closed-loop control for machine communication. Section 4 describes the system architecture and process flow for automated stability testing in machining. Section 5 shows the

results for two tools tested on a commercial CNC machine tool. Results show convergence to the optimal stable parameter in fourteen tests. Section 6 presents a discussion on the practical implementation of the closed-loop control system for stability testing. An alternative test strategy that minimizes the risk of tool damage due to unstable results in testing and methods for implementation in different controllers are described. Conclusions and future work are described in Section 7.

2. Bayesian machine learning for stability identification and strategy for test parameter selection

This section describes the Bayesian machine learning method for stability map identification and the strategy for test parameter selection. The objective of the Bayesian machine learning method is probabilistic modeling of stability over the spindle speed-axial depth of cut domain given test results (stable or unstable). The Bayesian learning method enables learning the stability behavior with or without information on the structural dynamics and cutting force model coefficients using a small number of test points. Traditional machine learning methods, such as neural network or support vector machines, have two main challenges for stability identification in milling. First, machine learning models are purely data-driven and agnostic to the physical laws governing the process, resulting in predictions that may violate the operational constraints or physical laws for the process. Second, machine learning models require many data points to generalize beyond the training dataset and predict the stability map accurately. With the Bayesian learning method, knowledge of the process physics can be incorporated into the Bayesian learning framework. This enables the Bayesian learning method to converge to the stability boundary using a small number of test data points.

The Bayesian machine learning method for stability boundary identification is as follows. Before any tests are performed, a probability of stability map is generated in the spindle speed-axial depth of cut domain (denoted as the prior). This is done by discretizing the axial depth and spindle speed ranges into a grid of points and assigning a probability of stability to each point. If the information on the structural dynamics and the cutting force model is not available, the probability of stability is decided from the knowledge that it is more likely to observe an unstable result at higher axial depths of cut. After an experimental result is made available, the posterior probability of stability is given by the Bayes' theorem as shown in Eq. (1) [15]:
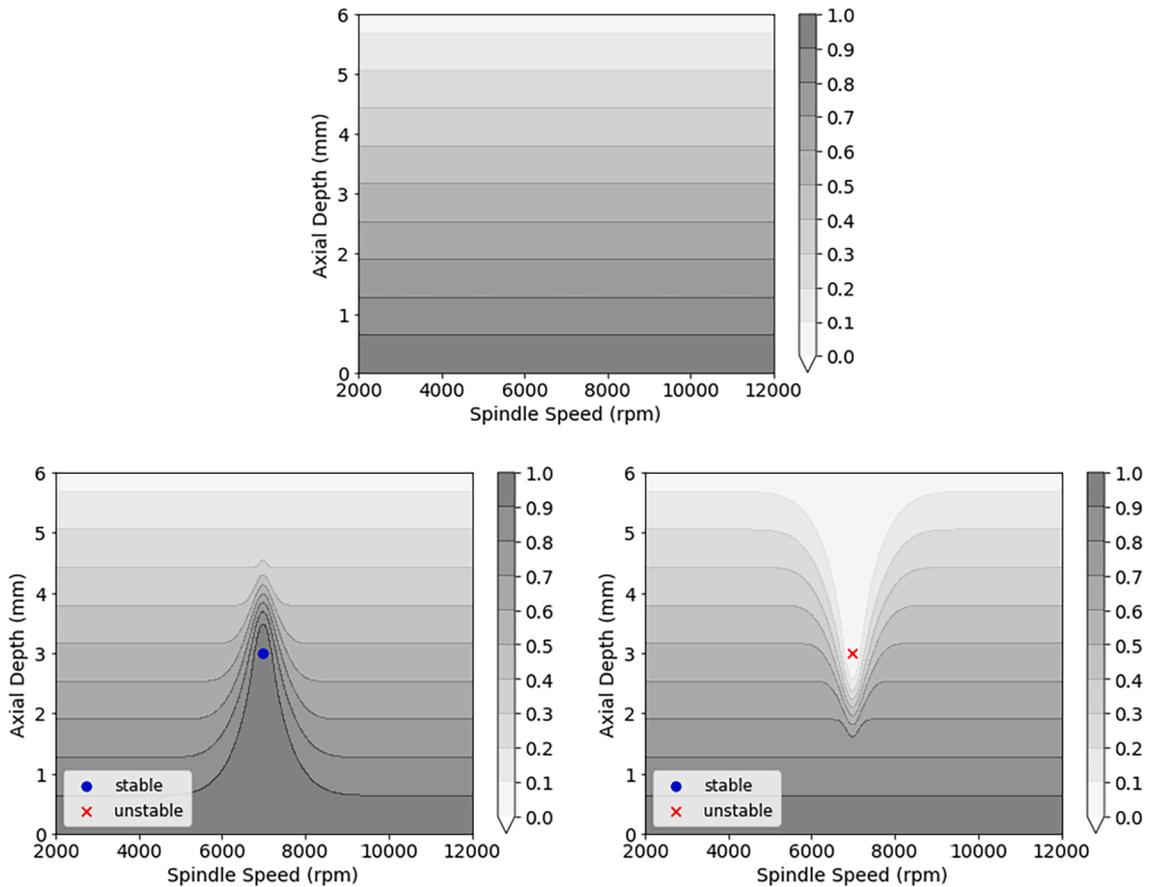
Fig. 1. Prior probability of stability (top panel) and the posterior probability of stability given stable result (bottom left panel) and an unstable result (bottom right panel) at {6500 rpm, 5 mm}.

$$p\left(s_g \mid r_t\right) = \frac{p\left(r_t \mid s_g\right) p\left(s_g\right)}{p\left(r_t\right)} \tag{1}$$

In Eq. (1), p denotes probability, s denotes stability, r denotes the test result, subscript g denotes an arbitrary grid point, and subscript t is the test point. The test result at grid point t can be either stable or unstable. According to the Bayes' rule, the posterior probability of stability at any point g given experimental result at t, denoted by $p(s_g \mid r_t)$, is the product of the prior probability of stability at g, $p(s_g)$, and the likelihood probability of experimental result given g is stable, $p(r_t \mid s_g)$, divided by the probability of the result, $p(r_t)$. Eq. (1) can be used for one or multiple test results at different test parameters in the domain. For multiple experimental results, the posterior probability of stability after the first experimental result becomes the prior for the second result and so on. Alternatively, multiple experimental results can be used simultaneously to calculate the posterior by taking a product of the likelihood probabilities for each experimental result. The physics and knowledge of the process can be encoded in the Bayes' learning algorithm through the prior and the likelihood probabilities. For brevity, the mathematical details of the Bayesian learning algorithm are not included in the paper; the reader is referred to [15] for details.

To illustrate the Bayesian learning method, consider a spindle speed range from 2000 rpm to 12000 rpm and an axial depth of cut range from 0 mm to 6 mm. Fig. 1 shows the prior probability of stability (top panel) and the updated probability of stability given stable result (bottom left panel) and an unstable result (bottom right panel) at {7000 rpm, 3 mm}. In this case, the prior probability of stability is taken to decrease linearly from one at the minimum axial depth of cut to 0.05 at the maximum axial depth of cut. As noted, the physics of the stability diagram is also encoded in the likelihood probabilities. For example, if a test result is stable, all axial depths of cut greater than the test axial depth of cut at the test spindle speed are also stable. The reader is referred to [15] for details on including the stability diagram knowledge in the likelihood probabilities. As seen from Fig. 1, one test result enables updating the probability of stability over a range of spindle speeds and axial depths, not just the test point. As noted, this is the advantage of Bayesian learning over traditional machine learning approaches, such as neural networks or support vector machines, which require large datasets to establish input-output correlations.

The optimal test parameter is decided based on an expected improvement in material removal rate, MRR, criterion [15]. The expected improvement in MRR at a grid point is given by Eq. (2).

$$\mathbb{E}[I(MRR)]_g = p\left(s_g\right) \times \frac{\left(MRR_g - MRR_{prior}\right)}{MRR_{prior}} \tag{2}$$

In Eq. (2), $\mathbb{E}$ denotes expectation, I denotes improvement, subscript g denotes an arbitrary grid point, MRR_{prior} is the optimal MRR based on the prior probability of stability, and MRR_g denotes the MRR at the grid point g. MRR_{prior} is determined as the highest MRR among parameters that are stable with certainty in the domain. The $\mathbb{E}[I(MRR)]$ criterion balances the trade-off between the probability of stability at a grid point and the improvement in MRR if the result at the grid point is stable. Fig. 2 shows $\mathbb{E}[I(MRR)]$ at all spindle speed-axial depth grid points given the linear prior probability shown in Fig. 1 (top panel); the color bar shows $\mathbb{E}[I(MRR)]$ as percentages. The optimal test point is {12,000 rpm, 3.2 mm}. Since no stable operating points are known before testing, the prior optimal point is taken as {12000 rpm, 0.01 mm}.

Fig. 3 illustrates the Bayesian learning method and the sequence of tests using $\mathbb{E}[I(MRR)]$ criterion. In Fig. 3, the test result is determined using the analytical stability boundary shown in green. The left column shows $\mathbb{E}[I(MRR)]$ for each test and the right column shows the posterior probability of stability using the test results. The sequence of tests is shown on the posterior probabilities of stability. The updated probability of stability after each test is used to calculate the test parameter for the next test using the expected improvement in MRR shown in Eq. (2). If the test result is stable, the MRR_{prior} is updated to the MRR at the stable test parameters. As seen from Fig. 3, the algorithm converges to the optimal stable parameters in four tests for the analytical stability boundary (shown in green). The testing can be terminated when the $\mathbb{E}[I(MRR)]$ is less than 5%. As shown in [10], convergence to the optimal stable parameters is typically achieved in 10 to 20 tests with the expected improvement in the MRR criterion.
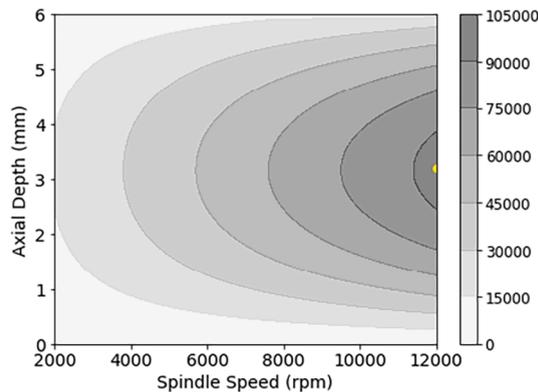
Fig. 2. Expected percentage improvement in MRR; the optimal test point is {12,000 rpm, 3.2 mm }.
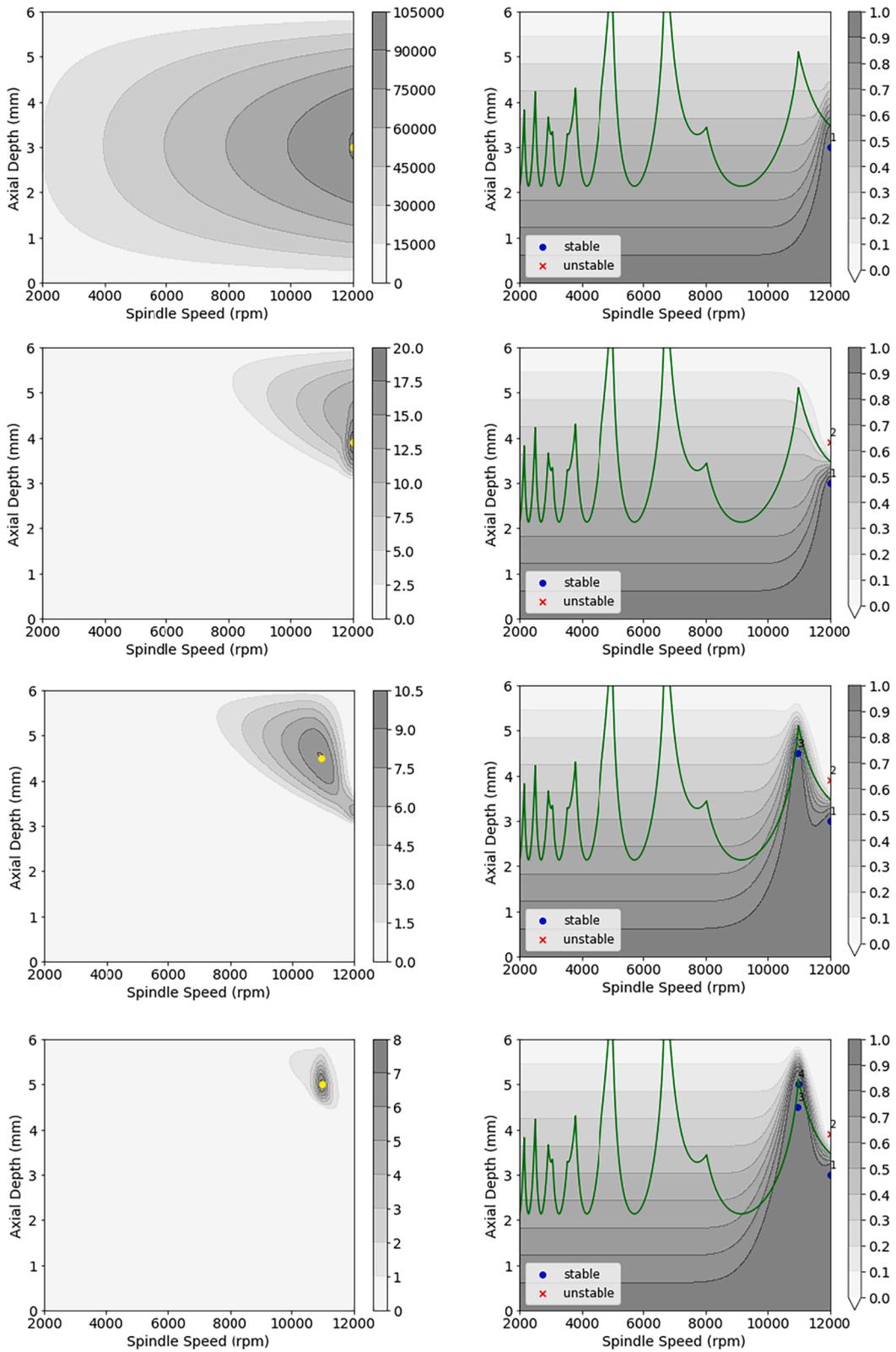
Fig. 3. Numerical demonstration of the test sequence and the Bayesian learning method based on the analytical stability boundary prediction; the left column shows E[$I(MRR)$] as percentages for each test and the right column shows the posterior probability of stability using the test result.

3. Closed-loop control for machine communication

This section describes the closed-loop control mechanism for machine communication. The closed-loop system consists of three parts: 1. an architecture to capture, communicate, and store information about the process; 2. an analysis framework to compare current actions against the desired actions and compute corrective steps; and 3. a feedback mechanism to provide updates to the machine instructions [16,17].

The process monitoring architecture is implemented on a standard PC-based computer next to the CNC machine. The computer is connected to the CNC controller via Ethernet and communicates using the TCP/IP protocol, providing a pathway for information output and input. MTConnect and OPC-UA are the two common protocols for capturing machine information [34–35]. The protocols provide information from the machine controller, such as the machine state, current machine position, feed rate, spindle speed, and part program line number [34–35]. The MTConnect protocol defines a common formatting standard for the information, which increases flexibility over multiple machines and controllers. A support framework captures process data generated by the CNC machine and provided via an XML file through MTConnect. The desired process tags such as machine state, machine position, and process parameters are parsed from the process data [36]. Note that MTConnect is a one-way protocol by design, preventing any information from being transferred to the CNC system [34]. OPC-UA was designed as a bi-directional communication protocol, enabling clients to provide information back to the server. However, most commercial CNC systems prevent bi-directionality and only allow information output, like the functionality provided by MTConnect. OPC-UA and MTConnect are interchangeable and achieve the same machine monitoring purpose. Each of these process tags is converted into a local nomenclature for comparison across multiple machining systems. This enables the simultaneous storage of all process data for external analysis and immediate processing of selected variables to inform the feedback mechanism. In addition to the data from the machine controller, external sensors can be retrofitted to the machine. The choice of the external sensor depends on the specific needs of the manufacturing process. In machining, the common external sensors include current and power sensors, accelerometers, and microphones. Like MTConnect and OPC-UA protocols, sensor data is converted into a local format for processing. Information from the machine protocols can be synchronized with the external sensors in the support framework.

The analysis framework governs the type of data collected by the process monitoring framework. The analysis framework is responsible for managing communications to and from the machine, triggering calculations for the next set of machine instructions, and triggering the feedback mechanism to communicate the instructions to the CNC controller. Multiple processes are required for analysis, depending on the type, amount, and format of data collected. The analysis framework relies on a common JSON data structure, with messages sent via Message Queue Telemetry Transport (MQTT) [36]. The analysis can be completed on a standard PC-based computer as minimal bandwidth is required. A new set of instructions for the machine are generated using the monitoring data from the machine, which are communicated to the machine using the feedback mechanism. All information transferred between the CNC controller and PC is text-based and requires less than 30 Kb per MTConnect message and less than 20 Kb per machine instruction message. While this system was implemented with wired connections, it is easily transferrable to wireless protocol and other local area networks.

The feedback mechanism enables timed updates to the current instructions on the CNC machining system. A series of flags are used between the CNC machine and the process monitoring architecture to determine when additional commands can be sent, following the timing sequence illustrated in Fig. 4. Fig. 4 shows a time of 2000 ms to transfer new instructions to the machine; this is defined by the user. Once the user enables the feedback process, the flag is set to high (1) and it signals that additional commands can be sent, overwriting previous commands in the specified memory location. The CNC machine then signals that no more instructions can be written by setting the flag low (0). Finally, the updated instructions are parsed by the machine, and the motion paths are executed. The CNC machine resets the flag to high (1) after the motion paths are executed indicating that it is ready for the next set of instructions. Fig. 5 displays the generic operational flow for the feedback mechanism. This flow is designed to work on any commercial CNC system with a standard operating system.
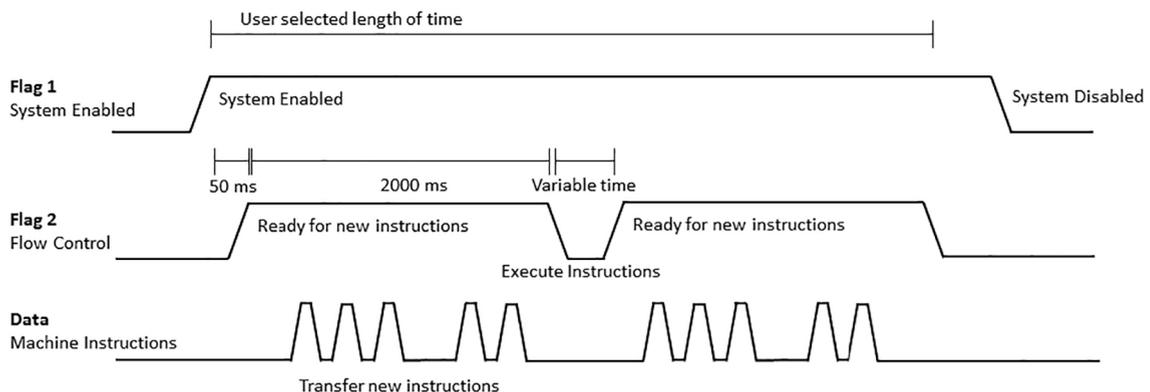
Fig. 4. Timing diagram for the feedback mechanism to update machine instructions.
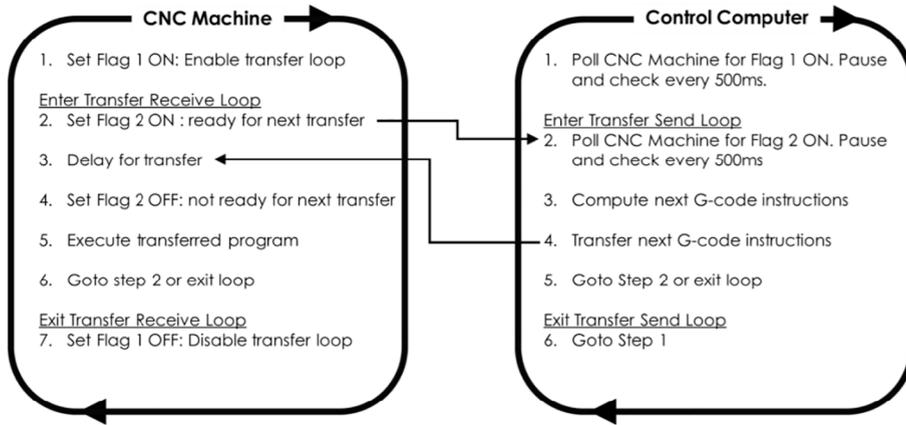
Fig. 5. Generic operational flow for the feedback mechanism.

4. System for automated stability testing with closed-loop control

This section describes the application of Bayesian learning, described in Section 2.0, and closed-loop control, described in Section 3.0, for automated stability testing. Fig. 6 shows the closed-loop control system flowchart. The user inputs information on the material for testing, tool, and process parameters. The tests are performed on a rectangular block of material. The inputs for the block include the maximum allowable cut length, width, and depth for testing and the material type. Tool information includes the type of tool (square, ball nose, or bull nose), tool diameter, number of teeth, helix angle, and pitch type (fixed or variable). Process parameter information includes the axial depth and spindle speed ranges, feed per tooth, radial depth of cut, and the type of cut (up milling or down milling). The user also inputs the stable operating parameters, if are known. The system proceeds as follows:

1. The Bayesian stability learning algorithm calculates the probability of stability in the spindle speed-axial depth of cut domain (the spindle speed and axial depth of cut ranges are discretized with a step size of 10 rpm and 0.1 mm, respectively);
2. the test parameter selection algorithm calculates the optimal test parameters (spindle speed and axial depth) to test based on the E[I (MRR)] criterion;
3. the G-code is updated with the selected test parameters and transferred to the machine;
4. the machine performs the test cut at the selected parameters;
5. an audio signal is recorded during the test and is used to classify the cut as stable or unstable;
6. steps 1 through 5 are repeated until E [I(MRR)] is less than 5%.

The test cuts proceed on the rectangular block of material as follows. The left-forward corner of the test block is set as the origin. The starting coordinate and the ending coordinate in X for the tool path are determined based on the type of cut, up or down milling, as noted by the user. The test cuts proceed using the axial depth of cut at the user-defined radial depth of cut, illustrated in Fig. 7. If the
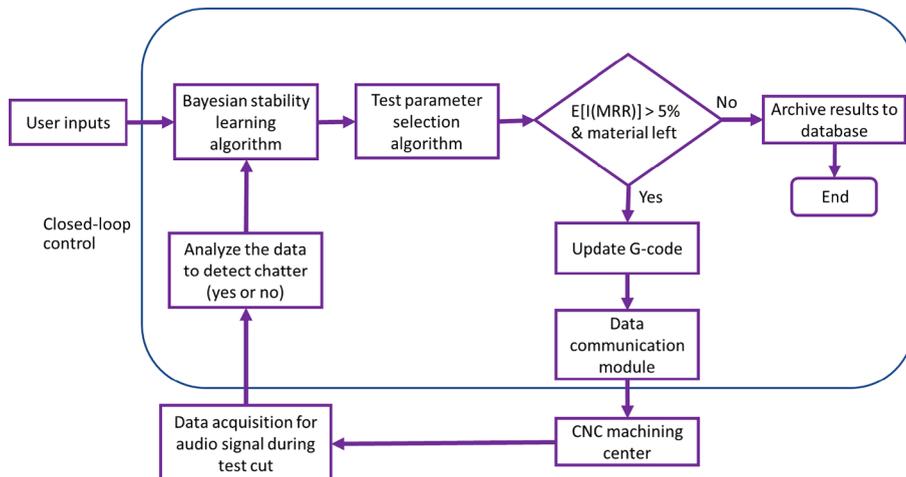
Fig. 6. Process flow for the automated stability testing system.
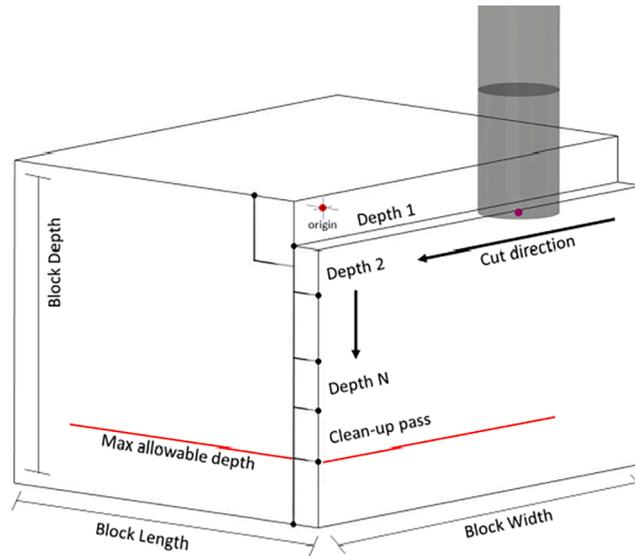
Fig. 7. Depth of cut progression across test block to determine machining stability. A clean-up pass is used if the selected test axial depth of cut exceeds the remaining available stock.

recommended test axial depth of cut is greater than the remaining block depth, a clean-up pass is performed at an axial depth of cut equal to the remaining block depth. The tool then increments in the radial depth of cut for the subsequent test. This is illustrated in Fig. 7.

5. Implementation and results

This section describes the implementation details on a Mazak VC-500A/5X AM HWD Hybrid machine. The Mazak had a Mazatrol SmoothX controller and operates on the Windows 7 platform. Table 1 lists the machining parameters (shown as user inputs in Fig. 6).

The machine state was monitored through the MTConnect protocol [34]. The machine was connected to a local network such that a secondary local computer could send and receive information. A machine state monitoring script was programmed in Python, enabling the local computer to receive MTConnect information produced by the machine via HTTP GET commands. Current machine state, machine position, user flags, and process parameters were parsed from an XML file and saved to the local machine at a frequency of 2 Hz. At the start of testing, the user flag was set to high (1), meaning the machine was ready for the next set of instructions. The flag value was monitored through the MTConnect protocol. The analysis framework, which consisted of the Bayesian machine learning and test parameter selection algorithms, was triggered when the flag value was high. The algorithms were implemented in Python. As noted, before any tests were performed, the Bayesian learning method started with an uninformed prior (shown in Fig. 1) based on the user-defined spindle speed and axial depth of cut range. The test parameter selection algorithm used the prior probability stability to calculate the optimal test parameters by Eq. (2).

Fig. 8 shows the prior probability of stability for the selected tool (left panel) and the E[*I*(*MRR*)] in percentage for the first test (right panel). The user input on block dimensions, process parameters, and the calculated test parameters were used to update the G-code. A G-code template for the Mazak machine was provided to the analysis framework as a starting point. The algorithm also updated the starting and ending X position of the tool based on the block length. The Y position was calculated using the radial depth of cut and the feed was calculated using the selected spindle speed, feed per tooth, and the number of teeth on the tool. The starting G-code template

Table 1
Machining parameters for testing.

#	Variable	Value
1	Workpiece material	Aluminum 6061-T6
2	Workpiece length	250 mm
3	Workpiece width	100 mm
4	Workpiece depth	15 mm
5	Tool diameter	12.7 mm
6	Number of teeth	4
7	Axial depth of cut range	0.01 mm to 10 mm
8	Spindle speed range	4000 rpm to 9000 rpm
9	Feed per tooth	0.06 mm/tooth
10	Radial depth of cut	5 mm
11	Type of cut	Down milling
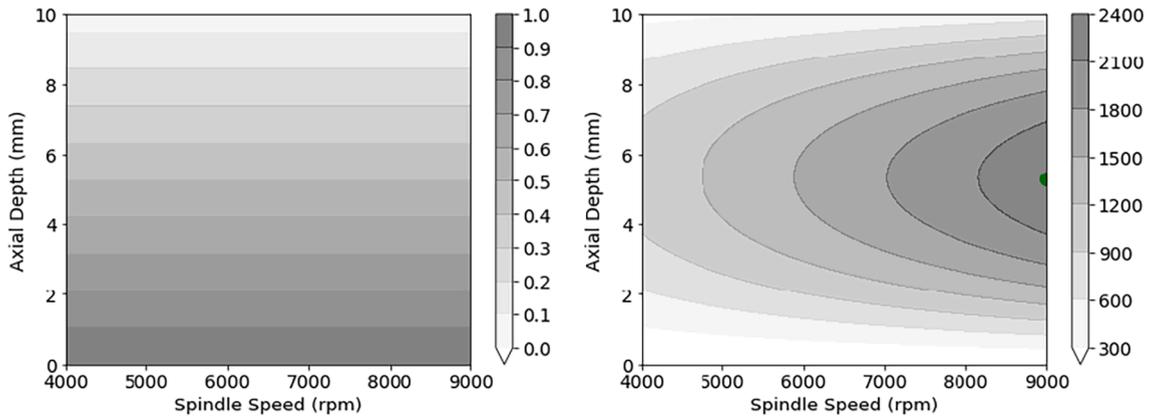
Fig. 8. Prior probability of stability for the selected tool (left panel) and the E[*I*(*MRR*)] in percentage for the first test (right panel).

and the updated G-code are shown in Fig. 9. In Fig. 9, the spindle speed is updated in line number N7, the X positions are updated in line number N9 and N12, the feed is updated in N9, the Y position is updated in N10, and the axial depth of cut is updated in N11.

As noted, the underlying operating system of the Mazak controller is Windows. This enables various memory locations internal and external to the controller to be mapped and networked. This provides a pathway to feed information back into the machine. The primary hard drive of the controller was networked to the local computer such that the computer could read and write files in a specific directory. Furthermore, current values of macro variables can be changed via G-code commands during live operations. The primary hard drive of the controller was networked to the local computer such that the computer could read and write files in a specific directory. This enabled a connected feedback loop, where the local computer could read the current status of the Machine via MTConnect and transfer the updated G-code to the machine via the networked hard drive.

Two G-code programs were used to operate the CNC machine, a primary control program, and a secondary sub-program. The user is responsible for initially executing the primary control program. This program sets the appropriate macro variables to signal that the machine tool had entered closed-loop control mode. The local computer polled the macro variable's status at a rate of 2 Hz. A variable delay was used to provide computation time for the local computer to analyze current data, generate the next set of instructions, and transfer them to the machine. Once ready, the local computer transferred a new set of instructions to the networked memory location on the Mazak controller's hard drive, coinciding with the secondary sub-program location. This new set of instructions can include any commands enabled by standard G-code, such as machine setup, toolpath, or inspection operations. After the delay expired, the controller entered the sub-program and executed the G-code statements. Finally, the sub-program returned to the main program, and the flag was reset to indicate that updated instructions were requested. This loop completes until a termination signal is given by the

```
N1  (--- pgm_start.txt ---)          N1  (--- pgm_start.txt ---)
N2 (STABILITY_TESTING)               N2 (STABILITY_TESTING)
N4 G28 G90 X0. Y0. Z0. B0. C0.       N4 G28 G90 X0. Y0. Z0. B0. C0.
N5 G53 G0 Z0.                        N5 G53 G0 Z0.
N6 T08 M06                           N6 T08 M06
N4 G54 G90                           N4 G54 G90
N7 S0000 M3                          N7 S9000 M3
N8 G0 Z10.0                          N8 G0 Z10.0
N9 G1 X-0.0 F0.0                      N9 G1 X275.4 F2160.0
N10 Y-0.0                            N10 Y-1.35
N11 Z-0.0                            N11 Z-5.3
N12 X0.0                             N12 X-25.4
N13 Z20.0                            N13 Z20.0
N14 M5                               N14 M5
N15 M9                               N15 M9
N16 G53 Z0.                          N16 G53 Z0.
N17 M30                              N17 M30
```

Fig. 9. G-code template (left panel) and the updated G-code with the selected axial depth of cut and spindle speed (right panel).

local computer. A common local HDF5 structure was used for data storage. This dataframe recorded machine state, actions, and control signals sent by the algorithm. At the end of each run, the dataframe was stored in an HDF5 format file, with a new table entry for each iteration of the stability algorithm. It should be noted that this particular choice of format was made for convenience with numerical analysis in Python NumPy and Pandas libraries. The specific implementation can also support other common formats such as SQL.

The procedure for automated stability testing with closed-loop control is as follows: The user first enables the closed-loop control process on the machine and the supporting computer. The CNC machine sets a system flag high (1) to indicate that it is ready for the next set of instructions. The machine is monitoring the system flags via MTConnect to determine when it is safe to write to the specified memory location. When the CNC machine flag is detected as high (1), it triggers the calculations to determine the optimal test parameters and update the G-code. After the flow control time flag time expires on the machine, the CNC machine sets the flag low (0), indicating that it is executing the G-code to perform the test cut and preventing the transfer of additional information from the local computer. The flow control flag time is user-defined and was set as 2000 ms as shown in Fig. 1.

Once a test is complete, the machine returns to the home position, and the flow control flag is reset to high (1), triggering the next set of calculations and notifying that the CNC machine is ready for the updated G-code. The analysis updates the probability of stability using the test result (stable or unstable) provided by the operator with the Bayesian machine learning method, calculates the test parameters for the next test and updates the G-code. The updated G-code with the next set of parameters is uploaded to the machine's memory and the loop repeats until $E[I(MRR)]$ is less than 5% or if the test material is exhausted. The flowchart for the system is shown in Fig. 6. An audio signal was recorded during the test cut and was used to classify the test cut as stable or unstable. The microphone signal was sampled at 20KHz during the cut. The Fast Fourier Transform (FFT) of the time-based audio signal collected during the test cut was calculated and the resulting spectrum was comb filtered to remove the tooth passing frequency and higher harmonics [37]. The dominant frequency was identified from the filtered spectrum. The cut was classified as unstable if the amplitude of the dominant frequency exceeded a certain threshold value. The threshold value is calibrated based on the background noise during the test process. Note that in this study, the microphone was not integrated into the closed-loop control system. The analysis to classify the test cut as stable or unstable was completed by the operator using the recorded signal and the result was provided to the system. However, the process monitoring framework enables the integration of the audio signal and the associated analysis for a completely automated system.

Fig. 10 shows the results. As noted, the tests were terminated when $E[I(MRR)]$ was less than 5%. The optimal point after 10 tests was {8710 rpm, 6.3 mm }. To validate the result, the FRF of the tools was measured using an impact test and an analytical prediction was made. The specific cutting energy (which represents the cutting force model) for the analytical stability prediction was 600 N/mm^2. The analytical prediction for the tool is shown in green Fig. 10. As seen from Fig. 10, the system was able to converge to the optimal stable parameters in 10 tests.

The procedure was repeated for a 9.525 mm four-tooth endmill. The spindle speed and axial depth of cut ranges were 4000 rpm to 9000 rpm and 0 mm to 5 mm, respectively. The block dimensions, feed per tooth, radial depth of cut, and the type of cut are listed in Table 1. Fig. 11 shows the results. The analytical prediction using the FRF measurement is shown in green. The testing had to be stopped and the tool had to be replaced after Test #8 and Test #14 due to excessive built-up edge on the tool. As seen from Fig. 11, the stability boundary was at a much lower axial depths cut compared to the selected maximum axial depth of cut. This resulted in many unstable tests as seen in Fig. 11.

6. Discussion

In this section, alternative algorithms for test parameter selection to reduce the risk of testing in the automating system are described.
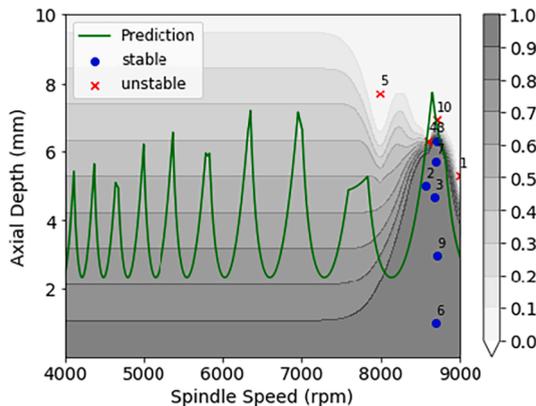
Fig. 10. Sequence of tests and posterior probability of stability for 12.7 mm diameter tool; the optimal stable point was {8710 rpm, 6.3 mm }. The analytical prediction is shown in green.
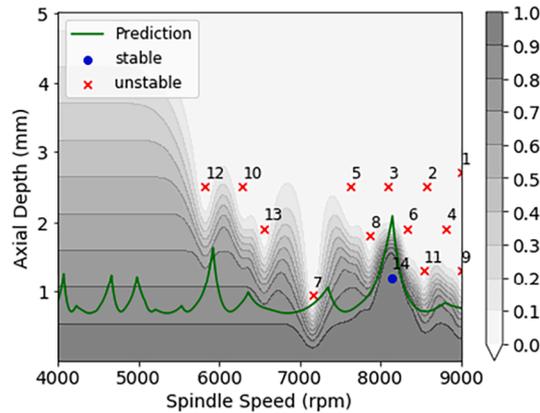
Fig. 11. Sequence of tests and posterior probability of stability for 9.525 mm diameter tool; the optimal stable point was {8150 rpm, 1.6 mm }. The analytical prediction is shown in green.

6.1. Algorithms for test parameter selection

As described in Section 2.0, an expected improvement in MRR criterion, denoted as E[*I*(*MRR*)], is used to select optimal test parameters. Although the E[*I*(*MRR*)] leads to faster convergence to the optimal stable parameters, it also results in many unstable tests in cases where the optimal stable axial depth of cut is a small fraction of the selected maximum value for the axial depth of cut. As seen from Fig. 2 and Fig. 8, the linear prior probabilities result in the first test at approximately 52% of the maximum axial depth of cut. This is due to a product of the linearly decreasing prior probabilities of stability and linearly increasing MRR as a function of axial depth of cut. This presents a challenge in automating the testing of tools; the resulting vibrations and forces from an unstable cut can damage the tool (and potentially the tool holder and spindle) and disrupt the automated testing process. To prevent this, two algorithms for test point selection are evaluated. First, the prior probabilities were decreased non-linearly using an exponential decay function. This reduces the axial depth of cut for the first test. Fig. 12 (left panel) shows the prior probability of stability using non-linear reduction in probabilities with axial depth of cut for the 12.7 mm diameter tool. Fig. 12 (right panel) shows a comparison between linear and non-linear reduction in prior probabilities as a function of axial depth of cut.

The two tools from experimental testing were studied numerically using the non-linear prior probabilities shown in Fig. 12. Fig. 13 displays the results for the 12.7 mm diameter tool (left panel) and the 9.525 mm diameter tool (right panel). Note that the results shown in Fig. 13 are numerical and were determined using the analytical stability boundary shown in green. As seen from Fig. 13 (right panel), although the first test axial depth of cut is lower, the algorithm doesn't show improvement in eliminating the unstable test cuts for the 9.525 mm diameter tool. This is because once a stable result is identified, the algorithm identifies successively higher axial depths of cut which have a higher MRR than the stable test result. Note that test cuts #10 and #24 for the 9.525 mm diameter tool shown in Fig. 13 (right panel) are clean-up passes due to the limited test block depth.

As noted, there are two main issues with the E[*I*(*MRR*)] criterion for automating test parameter selection. First, if the optimal stable axial depth of cut is a small fraction of the maximum axial depth of cut, the criterion will result in many unstable cuts before the optimal parameters are identified (see Fig. 11). This can damage the tool leading to disruptions in testing. Second, after a stable result is identified, the algorithm identifies successively higher axial depths of cut which have a higher MRR than the stable test result (see Fig. 13 right panel). To overcome these two issues, a modified test selection criterion based on a minimum probability of stability at the test parameters was evaluated. In this criterion, if the test parameter selection using the E[*I*(*MRR*)] has a probability of stability less than a pre-determined threshold probability, the axial depth of cut is reduced at the selected spindle speed such that the probability of stability equals the selected threshold probability. Fig. 14 shows the results for the two tools with a threshold probability equal to 0.8. The prior probability was linear.

As seen in Fig. 14, the modified criterion with a threshold probability reduces the number of unstable cuts and eliminates unstable test cuts that may damage the tool. The limitation of the method is that the number of tests required to converge can be higher if the optimal stable axial depth of cut is comparable to the maximum axial depth of cut. This is seen in Fig. 14 (left panel) for the 12.7 mm diameter tool where the number of tests required to converge increased from 10 (see Fig. 10) to 16. However, note that the modified criterion eliminates test cuts that can be excessively unstable (for example, test #5 in Fig. 10) that can damage the tool. This makes the modified criterion suitable for implementation in an automated test setup with closed-loop control. Additional precautions may be taken to prevent damage to the tool, holder, or spindle during testing. One approach is to monitor the spindle power during test cuts. Expected spindle power can be calculated using the test parameters and the specific power of the workpiece. A comparison between the measured spindle power and the expected power can be used to detect excessive tool wear or tool breakage to prevent damage to the holder and spindle.
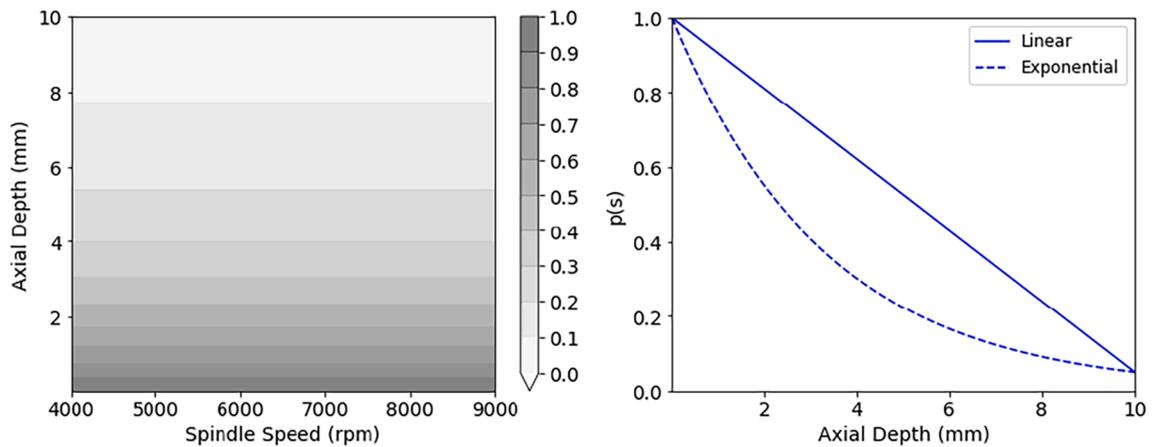
Fig. 12. Prior probability of stability using non-linear reduction in probabilities with axial depth of cut for the 12.7 mm diameter tool (left) and comparison between linear and non-linear reduction in prior probabilities as a function of axial depth of cut (right).
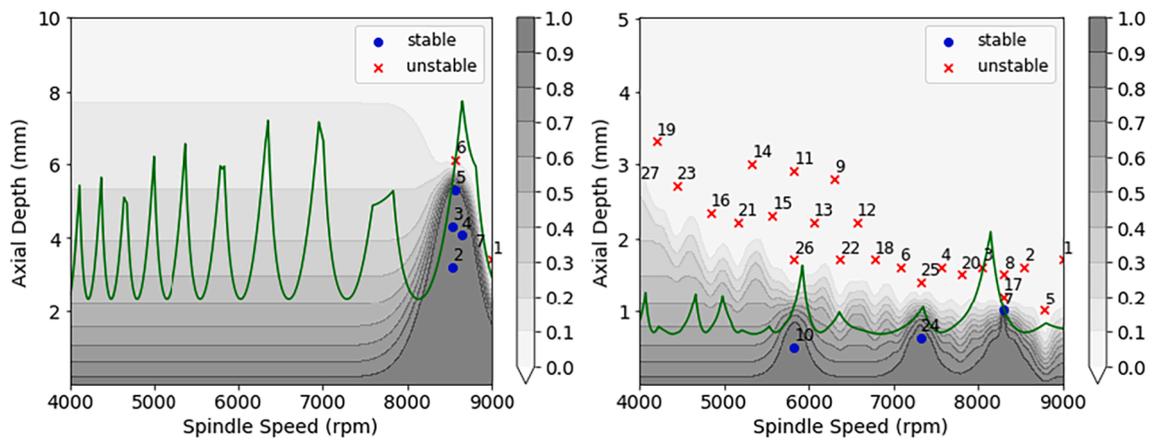
Fig. 13. Sequence of tests starting with a prior with a non-linear decrease in the probability of stability with the axial depth of cut for the 12.7 mm diameter tool (left panel) and 9.525 mm diameter tool (right panel).
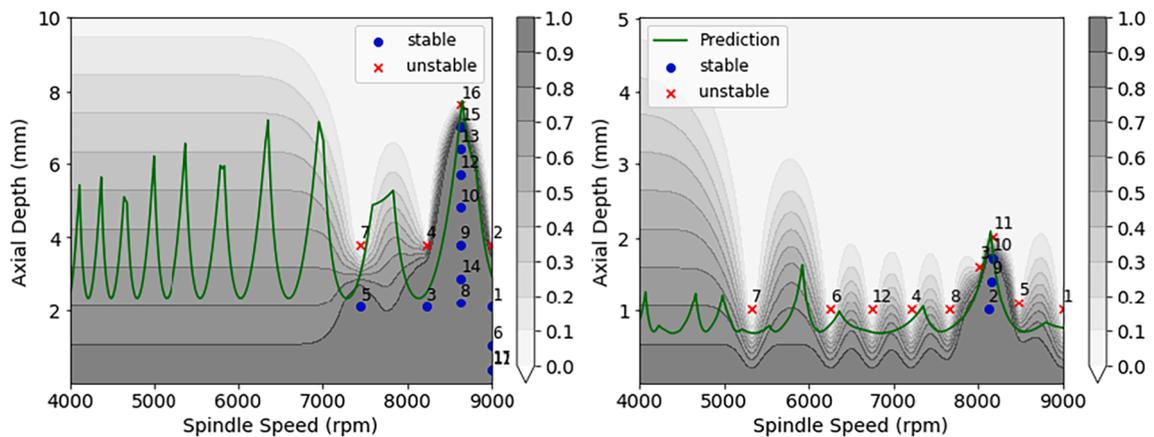
Fig. 14. The sequence of tests with a modified criterion of testing at an axial depth of cut with a probability of stability equal to or greater than 0.8 at the selected test spindle speed; the left panel shows the results for the 12.7 mm diameter tool and the right panel shows results for the 9.525 mm diameter tool.

6.2. Expansion to additional systems

The implementation of the closed-loop control system on a Mazak machine with a Windows-based Mazatrol SmoothX controller was described in Section 5.0. The implementation described in Section 5.0 relies on a data communication module to provide input instructions to the machine. This module is specific to the machine manufacturer, model, and machine controller. The methodology described in Section 5.0 involved the use of networked memory locations, managed by the data communication module, which applies to machining centers whose controllers run a version of the Windows operating system in the background. However, other methods of information transfer are possible enabling implementation on a range of commercial machining centers.

Well-supported controllers such as Siemens, Fanuc, and Linux CNC platforms retain the capability to accept instruction information through similar concepts. A secondary method of providing simple instructions relies on the "DPRNT" serial communication, or TCP/IP functionality in these controllers. When implemented in the data communication module, each line of G-code is sent and executed individually. This method is lower in bandwidth but is supported by most controllers. Additionally, other information transfer opportunities exist in the main program/sub-program structure described in Section 5.0. While the implementation described in Section 5.0 relied on cohesive blocks of instructions sent as a file (the entire sub-program), the G-code sub-program itself can be parameterized and persistently stored on the controller. In this approach, only the updated parameters are sent via macro variable update commands through the host controller's serial input. This methodology restricts the flexibility of the toolpath geometry but enables the rapid and efficient update of cutting depth, feed rate, and spindle speed. Additionally, this approach can be leveraged for many legacy controllers without modern networking capabilities.

7. Conclusions

A closed-loop system for automated identification of the optimal stable parameters in machining using Bayesian machine learning was described. The paper describes two new claims to the state of the art.

- A closed loop system consists of a process monitoring architecture, an analysis framework, and a feedback mechanism is described.
- The application of the closed-loop control for automated identification of the optimal stable parameters in machining using Bayesian machine learning is demonstrated.

The machine state and the test cuts were monitored through MTConnect and an audio signal. The Bayesian machine learning algorithm calculates the probability of stability at grid points given test results. The probability of stability was subsequently used to calculate the optimal test grid point which maximizes the expected improvement in material removal rate. The G-code is updated with the test parameters and communicated to the machine based on the machine status. The test cut is classified as stable or unstable based on the audio signal and the process is repeated until convergence to the optimal stable parameter is achieved. The closed-loop control system provides a low-cost and quick method for identifying optimal stable parameters in an industrial environment.

Future work will focus on algorithm improvements. First, the Bayesian learning approach can be extended to include the radial depth of cut to identify the optimal stable axial depth, radial depth, and spindle speed combination. Second, a low-cost microphone will be integrated into the process monitoring framework to enable a fully automated system. Additionally, the system will be evaluated on different machine tool controllers using the methods described in Section 6.2.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

References

[1] T. Schmitz, S. Smith, Machining Dynamics: Frequency Response to Improved Productivity, 2nd edn, Springer, New York, NY, 2019.
[2] J. Tlusty, Dynamics of High-Speed Milling, ASME J. Eng. Ind. 108 (2) (1986) 59–67.
[3] Y. Altintas, E. Budak, Analytical Prediction of Stability Lobes in Milling, Ann. CIRP 44 (1) (1995) 357–362.
[4] M.A. Davies, B. Dutterer, J.R. Pratt, A.J. Schaut, J.B. Bryan, On the Dynamics of High-Speed Milling With Long, Slender Endmills, Ann. CIRP 47 (1) (1998) 55–60.
[5] M.A. Davies, J.R. Pratt, B.S. Dutterer, T.J. Burns, The Stability of Low Radial Immersion Milling, Ann. CIRP 49 (1) (2000) 37–40.
[6] B.P. Mann, T. Insperger, P.V. Bayly, G. Stepan, Stability of Up-Milling and down-Milling—Part 1: Alternative Analytical Methods, Int. J. Mach. Tools Manuf. 43 (1) (2003) 25–34.
[7] T. Insperger, G. Stepan, P.V. Bayly, B.P. Mann, Multiple Chatter Frequencies in Milling Processes, J. Sound Vib. 262 (2) (2003) 333–345.
[8] D. Olvera, A. Elías-Zúñiga, H. Martínez-Alfaro, L.L. De Lacalle, C.A. Rodríguez, F.J. Campa, Determination of the stability lobes in milling operations based on homotopy and simulated annealing techniques, Mechatronics 24 (3) (2014) 177–185.
[9] F.I. Compean, D. Olvera, F.J. Campa, L.L. De Lacalle, A. Elias-Zuniga, C.A. Rodriguez, Characterization and stability analysis of a multivariable milling tool by the enhanced multistage homotopy perturbation method, Int. J. of Mach. Tools and Manuf. 57 (2012) 27–33.

[10] Z. Yan, Z. Liu, X. Wang, B. Liu, Z. Luo, D. Wang, Stability prediction of thin-walled workpiece made of Al7075 in milling based on shifted Chebyshev polynomials, The Int. J. of Adv. Manuf. Tech 87 (1) (2016) 115–124.

[11] C. Qin, J. Tao, H. Shi, D. Xiao, B. Li, C. Liu, A novel Chebyshev-wavelet-based approach for accurate and fast prediction of milling stability, Precision Eng 62 (2020) 244–255.

[12] H. Celikag, E. Ozturk, N.D. Sims, Can mode coupling chatter happen in milling? Int. J. of Mach. Tools and Manuf. 165 (2021) 103738.

[13] H.S. Kim, T.L. Schmitz, Bivariate uncertainty analysis for impact testing, Meas. Sci. and Tech 18 (11) (2007) 3565–3571.

[14] G. Duncan, T.L. Schmitz, M.H. Kurdi, Uncertainty propagation for selected analytical milling stability limit analyses, Trans. of the NAMRI/SME 34 (2000) 17–24.

[15] J. Karandikar, A. Honeycutt, T. Schmitz, S. Smith, Stability boundary and optimal operating parameter identification in milling using Bayesian learning, Journal of Manufacturing Processes 56 (2020) 1252–1262.

[16] Saleeby, K., Feldhausen, T., Kurfess, T., Love, L., 2020, System Level Control for Deposition Toolpaths in Hybrid Manufacturing, Int. Conf. on Additive Manuf, ASTM. November 16-20.

[17] Feldhausen, T., Saleeby, K., Kurfess, T., 2020, Spinning the Digital Thread with Hybrid Manufactruing, Manufacturing Letters, vol 29., pp. 15-18, 2021.

[18] S. Smith, J. Tlusty, Stabilizing chatter by automatic spindle speed regulation, CIRP Annals 41 (1) (1992) 433–436.

[19] Smith, S., Delio. T., 1992, Sensor-based chatter detection and avoidance by spindle speed, J. Dyn. Syst., Meas. Control, vol. 114, no. 3, 486–492, 1992.

[20] Delio, T., 1992, Method of Controlling Chatter in a Machine Tool, U.S. Patent No. 5,170,358.

[21] N.J.M. Van Dijk, E.J.J. Doppenberg, R.P.H. Faassen, N. Van De Wouw, J.A.J. Oosterling, H. Nijmeijer, Automatic in-process chatter avoidance in the high-speed milling process, J. of Dy. Sys., Meas., and Control. 132 (3) (2010).

[22] Y.S. Tarng, T.C. Li, The change of spindle speed for the avoidance of chatter in end milling, J. of Matl. Proc. Tech. 41 (2) (1994) 227–236.

[23] S.C. Lin, R.E. Devor, S.G. Kapoor, The effects of variable speed cutting on vibration control in face milling, ASME, J Eng Industry 112 (1990) 1–11.

[24] X. Long, B. Balachandran, Stability of up-milling and down-milling operations with variable spindle speed, J. of Vibration and Control 16 (7–8) (2010) 1151–1168.

[25] S. Seguy, G. Dessein, L. Arnaud, T. Insperger, Control of chatter by spindle speed variation in high-speed milling, Adv. Matl. Res. 112 (2010) 179–186.

[26] M. Zatarain, I. Bediaga, J. Munoa, R. Lizarralde, Stability of milling processes with continuous spindle speed variation: analysis in the frequency and time domains, and experimental correlation, CIRP Annals 57 (1) (2008) 379–384.

[27] L. Ding, Y. Sun, Z. Xiong, Online chatter suppression in turning by adaptive amplitude modulation of spindle speed variation, J. of Manuf. Sci. and Eng. 140 (12) (2018).

[28] L. Ding, Y. Sun, Z. Xiong, Active chatter suppression in turning by simultaneous adjustment of amplitude and frequency of spindle speed variation, J. of Manuf. Sci. and Eng. 142 (2) (2020).

[29] G. Urbikain, D. Olvera, L.L. de Lacalle, A. Elías-Zúñiga, Spindle speed variation technique in turning operations: Modeling and real implementation, J. of Sound and Vib. 383 (2016) 384–396.

[30] G. Totis, P. Albertelli, M. Sortino, M. Monno, Efficient evaluation of process stability in milling with spindle speed variation by using the Chebyshev collocation method, J. of Sound and Vib. 333 (3) (2014) 646–668.

[31] G. Urbikain, D. Olvera, L.N. López de Lacalle, A. Beranoagirre, A. Elías-Zuñiga, Prediction methods and experimental techniques for chatter avoidance in turning systems: A review, Appl. Sci. 9 (21) (2019) 4718.

[32] S. Schofield, P. Wright, Open Architecture Controllers for Machine Tools, Part 1: Design Principles, Journal of Manufacturing Science and Engineering 120 (2) (1998) 417–424.

[33] L.E.S. Oliveira, A.J. Álvares, Axiomatic Design Applied to the Development of a System for Monitoring and Teleoperation of a CNC Machine through the Internet, Procedia CIRP 53 (2016) 198–205.

[34] MTConnect Institute, 2018, MTConnect Standard.http://www.mtconnect.org/.

[35] Foundation, O., OPC Unified Architecture, https://opcfoundation.org/developertools/specifications-unified-architecture.

[36] U. Hunkeler, H.L. Truong, A. Stanford-Clark, Stanford-Clark, MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks, in: 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08), IEEE, 2008, pp. 791–798.

[37] W.R. Winfough, S. Smith, Automatic selection of the optimum metal removal conditions for high-speed milling, Trans. of NAMRI/SME 23 (1995) 163.